Matière : Développement Mobile

Auditoire : GLSI-3



TP4 Flutter

Routing – Communication entre les Widgets

NB

✓ Veuillez créer un nouveau Projet Flutter de type «Empty Application » et y ajouter le code des différentes activités de ce TP

Partie 1 : Navigation entre les Widgets

1- Créer sous le dossier « lib » les fichiers dart «**myApp** » , « **index** » « **chat**» et « **gererdiplomes** » contenant chacune la structure minimale d'un Stateless Widget. Compéter maintenant ces fichiers en suivant la description suivante :

(a) Fichier « **index** »

» L'interface contient un **TextButton** (c'est un texte cliquable). Le clic sur le bouton permet d'afficher l'interface ayant la route **/chat**

» Voici le code du TextButton

```
TextButton(
            onPressed: () { Navigator.of(context).pushNamed('/chat');},
            child: Text("Gerer Chat")
)
```

(b) Fichier « **myApp** »

» Dans sa fonction **build**, elle retourne un **MaterialApp. Y ajouter les routes de l'application** (associer un nom pour chaque widget). Ajouter les imports nécessaires

```
  Widget build(BuildContext context) {
    return MaterialApp(
      initialRoute: '/',
      routes : {
        '/' : (context) => Index(),
        '/chat' : (context) => Chat(),
      }
    );
  }
}
```

La route initiale remplace la propriété home

2- Ajouter ce qui est nécessaire dans **main.dart** afin de lancer le widget MyApp.

3- Ajouter l'icone home dans le AppBar du Widget « **Chat** ». Cette icone permet de retourner vers la page index. Cependant, l'icone ne possède pas un gestionnaire d'évènement prédéfini, nous devons donc soit utiliser le widget **IconButton** ou bien nous devons donc ajouter nous-même l'évènement correspondant à l'icône (c'est notre cas). Nous devons alors imbriquer ce Widget dans le widget **GestureDetector** (c'est un Widget qui détecte les évènements relatifs à son child)

```
appBar: AppBar(
            leading: GestureDetector(
                                    child: Icon(Icons.home),
                                    onTap: () {
                                                Navigator.of(context).pop();
                                                },
                                    )
        )
```



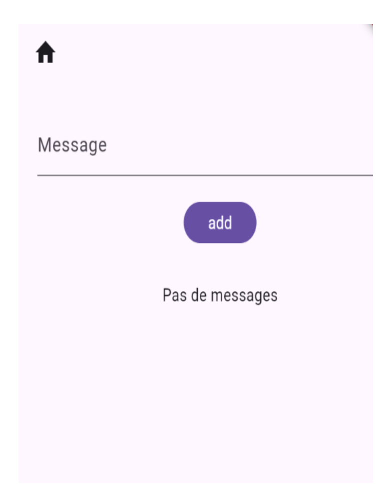
Exercice

- Ajouter ce qui est nécessaire afin d'avoir un autre lien dans le widget « index » vers « **gererdiplomes** »
- Vous pouvez ajouter le nom de la route dans le Widget correspondant sous la forme d'un attribut Static et pas le déclarer directement dans la liste des routes. Vous pouvez ainsi accéder au nom de la route à travers cet attribut. Refaire la route **gererdiplomes** selon cette idée.

Partie 2 : Communication entre Widgets Parent et Child

Activité 1 : Chat

Nous allons créer une simple interface dans laquelle nous enregistrons localement dans un **List<String>** des messages. L'interface affichera la liste des messages. Veuillez créer la structure du formulaire dans un Widget séparer.



1. Créer le widget **FormChat** contenant la structure de base du formulaire de chat (TextField + FilledButton).
2. Compléter le Widget Chat (de type Stateful) en ajoutant :
   a. Un attribut **List<String>messages** pour sauvegarder les messages saisis
   b. Une fonction **_addChat(String m)** : ajouteras le message « m » si non vide à la liste de message (mettre à jour l'état du widget).
   c. Dans la fonction build, retourner **le** FormChat et la liste de messages l'un à la suite de l'autre. Notez que vous devez utiliser une ListView pour afficher les messages et l'imbriquer dans le widget **Expanded** pour que la liste de message occupera le reste de l'interface.
   d. **Le FormChat** affichera le formulaire d'ajout du nouveau message et doit agir sur son Parent en appelant la fonction **_addChat(String m)** pour mettre à jour la liste des messages (**le fils doit agir sur les données de son parent**). Nous devons ainsi passer la fonction **_addChat(String m) en tant que fonction de Callback.**

Le voici le code du Widget FormChat

- **Pourquoi utiliser le Widget SingleChildScrollView??**) .
- Pourquoi utiliser un attribut de type TextEditingController est associé avec un TextField

```
class FormChat extends StatelessWidget {

  final chatController = TextEditingController(); //pour controller l'input
  final Function addChat; //fonction de callback

  FormChat(this.addChat);  //constructeur pour initialiser la fonction de callback

  @override
  Widget build(BuildContext context) {
    return SingleChildScrollView(
           child: Container(
            margin: const EdgeInsets.all(10),
            child: Column(children: [
                   TextField(
                              controller: chatController, //associer un controller
                              decoration: const InputDecoration(
                                                   labelText:'Message')),
                  SizedBox(height: 20),
                  FilledButton(
                               onPressed: () {
                                              //récupérer la Valeur de l'input
                                               String m = chatController.text;
                                               //Appel de la fonction de callback
                                                addChat(m);
                                               }
                                    child: const Text('add'))
            ])));
  }
}
```

Activité 2 : Gestion des diplômes

1- Créer une classe Diplôme (classe métier et pas une Interface) ayant comme attributs : numero, titre et mension. Y ajouter un constructeur avec des arguments nommés.
2- Créer les Widgets suivants :
   a. Widget « GererDiplomes.dart »
      i. contenant un attribut lesDiplomes (List<Diplôme>) contenant initialement deux diplomes initialisés de votre choix.
      ii. Composé de deux widgets situées l'une au-dessus de l'autre
      iii. Basé sur Scaffold
   b. Le premier widget contenant un formulaire pour ajouter un nouveau diplôme (utiliser le Widget Form) avec une fonction de callback
   c. Le deuxième qui affiche cette liste.
      i. Si pas de diplomes, afficher dans un Text « Pas encore de diplômes ».
      ii. Utiliser le Widget ListTile pour afficher les données d'un diplôme (numero et titre seulement) et y ajouter une icone pour afficher tous les details du dipolmes dans un Widget à part

